LISTING OF THE CLAIMS

Claims pending

- At time of the Action: Claims 1-22 and 25-31.
- After this Response: Claims 1-22 and 25-31.

Claims Canceled or Withdrawn Herein: 25-31.

Claims Amended Herein: 1, 12, 17 and 22.

New claims: None

1. **(Currently Amended)** A distributed information processing system, comprising:

a client device interface adapted to receive requests for electronic information from a plurality of remote devices;

a stateless module manager adapted to receive and route said requests from said client device interface; and

a plurality of information modules,

wherein said information modules register with said stateless module manager and stateless module manager routes said request to an appropriate one of said plurality of information modules in accordance with a type of information requested.

2. **(Original)** The distributed information processing system as recited in claim 1, wherein the requests to the device interface are formatted as an HTML or plain-text formatted e-mail.

3. **(Previously Presented)** The distributed information processing system as recited in claim 1, wherein the appropriate one of said plurality of information modules generates a response that is returned to said stateless module manager, and wherein said stateless module manager routes said response to said client interface device for delivery to a requestor.

4. **(Original)** The distributed information processing system as recited in claim 1, wherein said requests and responses are formatted as serializable Java objects.

5. **(Previously Presented)** The distributed information processing system as recited in claim 1, wherein said requests are made to said stateless module manager as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and returned when completed.

6. **(Previously Presented)** The distributed information processing system as recited in claim 1, wherein instances of said stateless module manager are created each time a new request is received and discarded after the request has been handled.

7. **(Previously Presented)** The distributed information processing system as recited in claim 6, wherein instances of said stateless module manager are stateless and multi-threaded.

8. **(Previously Presented)** The distributed information processing system as recited in claim 1, wherein information modules are loaded locally and remotely, wherein local modules reside on a same physical device as said stateless module manager, and wherein remote modules are located on other devices.

9. **(Previously Presented)** The distributed information processing system as recited in claim 8, wherein communication between locally loaded modules and said stateless module manager is accomplished via memory calls, object inheritance or inter-process communication.

10. **(Previously Presented)** The distributed information processing system as recited in claim 8, wherein communication between remotely loaded modules and said stateless module manager is accomplished via TCP/IP sockets.

11. **(Previously Presented)** The distributed information processing system as recited in claim 1, further comprising a subscription service that maintains a subscriber database, wherein information is sent by said information modules, and said subscriber database is consulted to determine to which clients the information should be forwarded.

12. **(Currently Amended)** A method of receiving and responding to requests for electronic information in a distributed information processing system, the method comprising:

receiving a request for electronic information at a client device interface;

forwarding said request to a stateless module manager;

consulting a registry of available information modules; and

forwarding said request to an appropriate information module as determined in accordance with a type of information requested.

13. **(Previously Presented)** The method of claim 12, further comprising:

maintaining a list of supported services provided by each of said information modules; and

handling service collisions if plural information modules are capable of responding to said type of information such that only one information module processes said request.

14. **(Original)** The method of claim 12, wherein said requests and responses are formatted as serializable Java objects.

15. **(Previously Presented)** The method of claim 12, wherein said requests are made to said stateless module manager as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and returned when completed.

16. **(Previously Presented)** The method of claim 12, said method further comprising:

creating an instance of said stateless module manager upon receiving said request; and

discarding said instance after said response has been handled.

17. **(Currently Amended)** A computer readable medium containing computer executable instructions for receiving and responding to requests for electronic information in a distributed information processing system, said computer executable instructions for performing the steps of:

receiving a request for electronic information at a client device interface;

forwarding said request to a stateless module manager;

consulting a registry of available information modules; and

forwarding said request to an appropriate information module as determined in accordance with a type of information requested.

18. **(Previously Presented)** The computer readable medium of claim 17, further comprising computer executable instructions for performing the steps of:

maintaining a list of supported services provided by each of said information modules; and

handling service collisions if plural information modules are capable of responding to said type of information such that only one information module processes said request.

19. **(Previously Presented)** The computer readable medium of claim 17, wherein said requests and responses are formatted as serializable Java objects.

20. **(Previously Presented)** The computer readable medium of claim 17, wherein said requests are made to said stateless module manager as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and returned when completed.

21. **(Previously Presented)** The computer readable medium of claim 17, further comprising executable instructions for performing the steps of:

creating an instance of said stateless module manager upon receiving said request; and

discarding said instance after said response has been handled.

22. **(Currently Amended)** A stateless module manager that manages a request for electronic information received at a mailbox, comprising:

a registry of information modules;

a module loading function for dynamically loading said information modules upon receipt of said request for electronic information, wherein said request is made as one of a serializable Java object, XML placed in an HTTP header, or an XML-RPC-enabled web server, wherein said request is either synchronous or asynchronous, wherein a synchronous request is handled on a first-

in-first-out basis, and wherein an asynchronous request is processed and a response returned in accordance with a processing time of the request;

wherein said stateless module manager routes said request to an appropriate information module for resolution, and wherein said appropriate information module resolves said request and returns a response to said stateless module manager;

wherein said stateless module manager maintains a list of supported services provided by each of said information modules and handles service collisions such that if plural information modules register as supporting a same service by determining which of said plural information modules will handle said request;

wherein instances of said stateless module manger are created each time a new request is received and discarded after the request has been handled;

wherein said stateless module loading function includes local and remote module loading functions, wherein said local loading function loads information modules that reside on a same physical device as said stateless module manager, wherein said remote loading function loads information modules that reside on devices logically connected to said stateless module manager, wherein said local modules communicate with said stateless module manager via one of memory calls, object inheritance, and inter-process communication, and wherein said remote information modules communicate with said stateless module manager via TCP/IP sockets; and

further comprising a user interface, wherein said user interface is adapted to configure said stateless module manager.

23. **(Cancelled)**

24. **(Cancelled)**

25. **(Cancelled).**

26. **(Cancelled).**

27. **(Cancelled).**

28. **(Cancelled).**

29. **(Cancelled).**

30. **(Cancelled).**

31. **(Cancelled).**